Regularized Extreme Learning Machine for Multi-view Semi-supervised Action Recognition

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, Greece

{*aiosif,tefas,pitas*}*@aiia.csd.auth.gr*

Abstract

In this paper, three novel classification algorithms aiming at (semi-)supervised action classification are proposed. Inspired by the effectiveness of discriminant subspace learning techniques and the fast and efficient Extreme Learning Machine (ELM) algorithm for Single-hidden Layer Feedforward Neural networks training, the ELM algorithm is extended by incorporating discrimination criteria in its optimization process, in order to enhance its classification performance. The proposed Discriminant ELM algorithm is extended, by incorporating proper regularization in its optimization process, in order to exploit information appearing in both labeled and unlabeled action instances. An iterative optimization scheme is proposed in order to address multi-view action classification. The proposed classification algorithms are evaluated on three publicly available action recognition databases providing state-of-the-art performance in all the cases.

Keywords: Extreme Learning Machine, Semi-supervised Learning, Multi-view Learning

1. Introduction

Human action recognition from videos is receiving increasing attention, due to its importance in a wide range of applications, like intelligent visual surveillance, human-computer interaction and content-based video annotation/retrieval, to name a few. However, it is a challenging problem, because of the complexity of human actions. The dynamic human body motion patterns can produce an extremely large number of visual representations, due to the large number of the

degrees of freedom in body joints, the differences in human body size, action execution style variations among individuals, differences in camera distance and view angle and imaging conditions changes, e.g. illumination changes. This fact leads to high intra-class and, possibly, moderate inter-class variations for human action classes.

The term action is often distinguished from the term activity. An action is referred to as a simple motion pattern [1], e.g., a walking step. Activities consist of a series of actions, e.g., the activity 'playing football' consists of successive realizations of actions 'run', 'jump', 'kick', etc. Therefore, each activity can be split into its elementary motion patterns called actions. Recent action recognition methods either exploit the local information appearing in video frame locations that correspond to space-time interest points (STIPs) [2], or divide each video depicting an action instance, called action video hereafter, in multiple short videos and describe each of the short videos in order to obtain a template-based action representation [3]. In the first case, shape and motion descriptors, like Histograms of Oriented Gradients (HOG) and Histograms of Optical Flow (HOF) are calculated, in order to describe actions. By employing such action descriptors, action videos are usually represented by adopting Bag of Features (BoFs) approaches [4]. In the later case, descriptors denoting the similarity of each sub-video with reference ones are calculated leading to a template-based action representation [3].

To achieve good recognition performance, most action recognition methods have approached the action recognition problem from a restricted, supervised, point of view. That is, they assume that actions are observed from one view angle only and that a large amount of labeled action videos are available in the training phase, in order to train action classifiers that will be used to classify unknown test action instances. This approach has been extensively studied in the last two decades, leading to high action classification rates in several action recognition datasets [5, 6, 7]. However, in real application scenarios, actions may be observed from various or multiple view angles, e.g., when using a multi-camera setup [8]. Furthermore, labeled action training samples are, usually, difficult or expensive to obtain, since they typically require manual annotation (tagging). Therefore, the achievement of a good learning model using a limited number of labeled action videos that can employ multiple visual representations for action instances is a crucial issue.

Despite the fact that action recognition has been extensively studied in the last two decades, there are few semi-supervised action recognition methods, which can exploit both labeled and unlabeled action videos in their training process. La-

beled Kernel Sparse Coding (LKSC) and l1 graphs are proposed in [9], in order to use unlabeled action videos in a sparsity-based action classification scheme. Semi-supervised discriminant analysis with global constraint (SDG) is proposed in [10]. SDG incorporates Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA) and Locality Preserving Projections (LPP) in one optimization scheme, in order to fuse the information appearing in both labeled and unlabeled action videos. Regarding multi-view action recognition, two approaches have been proposed. In the first one, multi-camera setups are employed [8]. Thus, multiple action representations, one for each of the resulting action videos, are obtained. In the latter one, action videos are represented by multiple descriptors [11]. In both [8, 11], action video classification based on Feedforward Artificial Neural Networks and action classification decision fusion based on Bayesian Learning have been proposed, in order to train classifiers in with multi-view action data.

Extreme Learning Machine (ELM) [12] is a, relatively, new algorithm for fast Single-hidden Layer Feedforward Neural (SLFN) networks training, requiring low human supervision. Conventional SLFN training algorithms require adjustment of the network weights and the bias values, using a parameter optimization approach, like gradient descent. However, gradient descent learning techniques are, generally, slow and may lead to local minima. In ELM, the input weights and the hidden layer bias values are randomly chosen, while the network output weights are analytically calculated. By using a sufficiently large number of hidden layer neurons, the ELM classification scheme can be thought of as being a non-linear mapping of the training data on a high-dimensional feature space, called ELM space hereafter, followed by linear data projection and classification. ELM not only tends to reach a small training error, but also a small norm of output weights, indicating good generalization performance [13]. ELM has been successfully applied to many classification problems, including human action recognition [14, 15, 16, 17].

In this paper, we propose a novel method aiming at addressing multi-view semi-supervised action video classification. That is, we assume that the training set is formed by l labeled and u ($u \gg l$) unlabeled action videos, each depicting an action instance, where each action instance may be represented by multiple representations (views). In order to handle the non-linear structure of action classes, caused by the above described high intra- and moderate inter-action class variations, three novel non-linear classification algorithms exploiting the available information appearing in both labeled and unlabeled data, having one or multiple views, are proposed. To this end, we build on the ELM algorithm and extend it

along three directions:

- Exploiting the fact that the ELM classification scheme can be considered to be a non-linear data mapping from the input space to the high-dimensional ELM space, followed by a linear mapping to a low-dimensional feature space determined by the network target vectors (called output space hereafter), we incorporate discrimination criteria [18] in the ELM optimization process. As will be shown, several (linear or non-linear) discrimination criteria can be incorporated in the ELM optimization process, without any modification in the proposed optimization scheme.

- A graph-based regularizer describing the action video similarities in the ELM space is incorporated in the ELM optimization process, n order to exploit the available information of the unlabeled training action videos. By adopting a graph-based regularization scheme in the ELM space, non-linear relationships between action videos can be described.

- In order to simultaneously exploit multiple action views, a weighted regularization-based iterative optimization scheme is proposed, where each view contributes to the optimization process according to its action discrimination ability.

The contributions of the paper are the following ones. A novel algorithm for semi-supervised SLFN networks training is proposed that incorporates discrimination criteria on the ELM optimization process and is able to exploit information appearing in both labeled and unlabeled data. A novel optimization scheme for multi-view semi-supervised SLFN networks training is proposed. Finally, the proposed algorithms are evaluated on action recognition, by exploiting two action representations, providing state-of-the-art performance on three publicly available databases.

The rest of the paper is structured as follows. Section 2 provides an overview of the recognition framework used in the proposed approach. Section 3 describes the proposed classification algorithms in detail. Experimental results evaluating their performance are illustrated in Section 4. Finally, conclusions are drawn in Section 5.

2. Problem Statement

Let $\mathcal{U}$ be a video database formed by action videos depicting N_I action instances, each belonging to one of the N_A action classes forming an action class

set $\mathcal{A}$. Let us assume that the action videos in the database are processed, in order to produce $N_T = N_I \cdot N_V$ action vectors $\mathbf{s}_i^v \in \mathbb{R}^{D_v}$, $i = 1, \ldots, N_I$, $v = 1, \ldots, N_V$, where N_V is the number of views involved in the action classification problem and D_v is the dimensionality of the feature space related to view v. Any vector-based action representation can be employed to this end, since we do not set any assumption on the nature of the action vectors $\mathbf{s}_i$. In this paper, we have employed the recently proposed Action Bank [3] and the 3DHarris STIP detector [2], followed by the calculation of HOG/HOF descriptors [4], for action video representation. In the case of action recognition employing a multi-camera setup, N_V is the number of cameras forming the adopted camera setup, while in the case where each action video is represented by multiple representations, N_V is the number of the adopted action representations. Clearly, $N_V = 1$ when human action recognition employs one camera and one representation for action video representation.

Let us also assume that the database is partially annotated. That is, let us assume that l of the N_I action instances are labeled with action class labels c_i, $i = 1, \ldots, l$, each belonging to one of the N_A action classes forming $\mathcal{A}$. We would like to employ both the $L = l \cdot N_V$ labeled and the $U = u \cdot N_V$, $u = N_I - l$ unlabeled action vectors, in order to train a classifier that can be used to classify any unknown test action instance represented by $N \leq N_V$ test action vectors $\mathbf{s}_t^v$, $v = 1, \ldots, N$.

3. Proposed Method

In this Section, we describe in detail three classification algorithms. We start with the description of a single-view supervised classification algorithm in Subsection 3.2. An extension of this algorithm to exploit both labeled and unlabeled data information is described in Subsection 3.3. Finally, an optimization scheme aiming at multi-view (semi-)supervised action classification is described in Subsection 3.4. Since, as has already been mentioned, the proposed classification algorithms are extensions of the ELM algorithm, we briefly overview it in Subsection 3.1.

3.1. Extreme Learning Machine

ELM has been proposed for single-view supervised classification [12]. In the following description, we drop the superscript denoting the view index, for notation simplicity. We will use it again in subsection 3.4, where we discuss multi-view action classification. Let $\mathbf{s}_i$ and c_i, $i = 1, ..., l$ be the set of the labeled action vectors and the corresponding action class labels, respectively. We would

like to employ them in order to train a SLFN network. For a classification problem involving the D-dimensional action vectors $\mathbf{s}_i$, each belonging to one of the N_A action classes, the network should contain D input, H hidden and N_A output neurons. The number of the network hidden layer neurons is, typically, chosen to be higher than the number of action classes, i.e., $H \gg N_A$. The network target vectors $\mathbf{t}_i = [t_{i1}, ..., t_{iN_A}]^T$, each corresponding to one labeled action vector $\mathbf{s}_i$, are set to $t_{ij} = 1$ for vectors belonging to action class j, i.e., when $c_i = j$, and to $t_{ij} = -1$ otherwise.

In ELM, the network input weights $\mathbf{W}_{in} \in \mathbb{R}^{D \times H}$ and the hidden layer bias values $\mathbf{b} \in \mathbb{R}^H$ are randomly chosen, while the output weights $\mathbf{W}_{out} \in \mathbb{R}^{H \times N_A}$ are analytically calculated. Let $\mathbf{v}_j$ denote the j-th column of $\mathbf{W}_{in}$, $\mathbf{u}_k$ the k-th row of $\mathbf{W}_{out}$ and u_{kj} be the j-th element of $\mathbf{u}_k$. For a given hidden layer activation function $\Phi()$ and by using a linear activation function for the output neurons, the output $\mathbf{o}_i = [o_1, \ldots, o_{N_A}]^T$ of the ELM network corresponding to training action vector $\mathbf{s}_i$ is given by:

$$o_{ik} = \sum_{j=1}^{H} u_{kj}\, \Phi(\mathbf{v}_j, b_j, \mathbf{s}_i), \quad k = 1, ..., N_A. \tag{1}$$

Many activation functions $\Phi()$ can be employed for the calculation of the hidden layer output, such as sigmoid, sine, Gaussian, hard-limiting and Radial Basis (RBF) functions. The most popular choices are the sigmoid and the RBF functions, i.e.:

$$\Phi_{sigmoid}(\mathbf{v}_j, b_j, \mathbf{s}_i) = \frac{1}{1 + e^{-(\mathbf{v}_j^T\, \mathbf{s}_i + b_j)}}, \tag{2}$$

$$\Phi_{RBF}(\mathbf{v}_j, b_j, \mathbf{s}_i) = e^{-b_j \|\mathbf{s}_i - \mathbf{v}_j\|_2^2}. \tag{3}$$

By storing the hidden layer neuron outputs in a matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi(\mathbf{v}_1, b_1, \mathbf{s}_1) & \cdots & \Phi(\mathbf{v}_1, b_1, \mathbf{s}_l) \\ \cdots & \ddots & \cdots \\ \Phi(\mathbf{v}_H, b_H, \mathbf{s}_1) & \cdots & \Phi(\mathbf{v}_H, b_H, \mathbf{s}_l) \end{bmatrix}, \tag{4}$$

equation (1) can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{\Phi}$. Finally, by assuming that the predicted network outputs $\mathbf{O}$ are equal to the desired ones, i.e., $\mathbf{o}_i = \mathbf{t}_i,\ i = 1, ..., l$, $\mathbf{W}_{out}$ can be analytically calculated by solving for:

$$\mathbf{W}_{out}^T \mathbf{\Phi} = \mathbf{T}, \tag{5}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_l]$ is a matrix containing the network target vectors. Using (5), the network output weights minimizing $\|\mathbf{W}_{out}^T \mathbf{\Phi} - \mathbf{T}\|_F$ are given by $\mathbf{W}_{out} = \mathbf{\Phi}^\dagger \mathbf{T}^T$, where $\|\mathbf{X}\|_F$ is the Frobenius norm of $\mathbf{X}$ and $\mathbf{\Phi}^\dagger = \left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1} \mathbf{\Phi}$ is the generalized pseudo-inverse of $\mathbf{\Phi}^T$.

After calculating the network output weights $\mathbf{W}_{out}$, a test action vector $\mathbf{s}_t$ can be introduced to the trained network and be classified to the action class corresponding to the maximal network output, i.e.:

$$c_t = arg \max_j o_{tj},\ j = 1, ..., N_A. \tag{6}$$

3.2. Discriminant Extreme Learning Machine

The ELM algorithm described above can be considered to be a non-liner mapping of the training action vectors $\mathbf{s}_i,\ i = 1, \ldots, l$ from the input space $\mathbb{R}^D$ to the high-dimensional ELM space $\mathbb{R}^H$, followed by a linear projection of the action vectors representation in the ELM space to the output space $\mathbb{R}^{N_A}$, determined by the network target vectors $\mathbf{t}_i$. From a Discriminant Analysis perspective, the calculation of the ELM output weights can be considered as the determination of an appropriate data projection matrix $\mathbf{W}_{out}$, used to map the ELM space to a low-dimensional feature space where action classes are better discriminated. Having this in mind, we propose to solve the following optimization problem for $\mathbf{W}_{out}$ calculation:

$$\mathcal{J}_1 = \frac{1}{2} Tr\left(\mathbf{W}_{out}^T \mathbf{S}_X \mathbf{W}_{out}\right) + \frac{\lambda_1}{2} \|\mathbf{W}_{out}^T \mathbf{\Phi} - \mathbf{T}\|_F^2, \tag{7}$$

where $Tr\left(\mathbf{X}\right)$ denotes the trace of $\mathbf{X}$, λ_1 is a parameter denoting the importance of the training error in the optimization problem and $\mathbf{S}_X$ is a matrix describing desirable properties of the training action vector set in the ELM space and is called discriminant matrix hereafter.

The first term in (7) has been widely used by discriminant analysis-based subspace learning techniques. For example, Principal Component Analysis (PCA) optimization process [19] employs the first term in (7) and the total scatter matrix:

$$\mathbf{S}_T = \sum_{i=1}^{l} \left(\boldsymbol{\phi}_i - \bar{\boldsymbol{\phi}}\right) \left(\boldsymbol{\phi}_i - \bar{\boldsymbol{\phi}}\right)^T, \tag{8}$$

in the place of $\mathbf{S}_X$ for the determination of a feature space for action vectors projection where the data variance is increased. Linear Discriminant Analysis

(LDA) [19] employs the within-class scatter matrix:

$$\mathbf{S}_w = \sum_{j=1}^{N_A} \sum_{i=1}^{l} \eta_i^j \left(\boldsymbol{\phi}_i - \bar{\boldsymbol{\phi}}_j\right) \left(\boldsymbol{\phi}_i - \bar{\boldsymbol{\phi}}_j\right)^T, \tag{9}$$

in the place of $\mathbf{S}_X$ for the determination of a low-dimensional feature space where action classes are better discriminated. In (8), (9), $\boldsymbol{\phi}_i$ is the i-th column of $\boldsymbol{\Phi}$, i.e., the representation of $\mathbf{s}_i$ in the ELM space, $\bar{\boldsymbol{\phi}} = \frac{1}{l} \sum_{i=1}^{l} \boldsymbol{\phi}_i$ is the mean action vector in the ELM space, while $\bar{\boldsymbol{\phi}}_j = \frac{1}{l_j} \sum_{i=1}^{l} \eta_i^j \boldsymbol{\phi}_i$ is the mean vector of action class j, having cardinality $l_j = \sum_{i=1}^{l} \eta_i^j$. η_i^j is an index denoting if action vector $\mathbf{s}_i$ belongs to action class j, i.e., $\eta_i^j = 1$ if $c_i = j$ and $\eta_i^j = 0$ otherwise. We should note here that the calculation of $\mathbf{S}$ in the ELM space $\mathbb{R}^H$, rather than in the input space $\mathbb{R}^D$, has the advantage that nonlinear relationships, which depend on the adopted activation function $\Phi()$, between training action vectors $\mathbf{s}_i$ can be better described. Furthermore, since action class discrimination in the projection space is handled by the second term in (7), $\mathbf{S}_X$ is chosen to describe action class compactness properties.

$\mathcal{J}_1$ in (7) is minimized by solving for $\frac{\theta \mathcal{J}_1}{\theta \mathbf{W}_{out}} = 0$. Then $\mathbf{W}_{out}$ is given by:

$$\mathbf{W}_{out} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{\lambda_1}\mathbf{S}_X\right)^{-1} \boldsymbol{\Phi}\mathbf{T}^T. \tag{10}$$

As has been shown in [18], a wide range of (linear and non-linear) discrimination criteria can be described from a graph embedding point of view. Let $\mathcal{G} = \{\boldsymbol{\Phi}, \mathbf{B}\}$ be an undirected weighted graph, where we assume that the training action vectors in the ELM space form the vertex set of the graph and $\mathbf{B} \in \mathbb{R}^{l \times l}$ is the corresponding graph adjacency matrix. The diagonal matrix $\mathbf{D} \in \mathbb{R}^{l \times l}$ and the graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{l \times l}$ are defined by $D_{ii} = \sum_{i \neq j} [\mathbf{B}]_{ij},\ i = 1, ..., l$ and $\mathbf{L} = \mathbf{D} - \mathbf{B}$, respectively. The graph Laplacian matrix $\mathbf{L}$ can be employed, in order to describe discrimination criteria exploited in several discriminant analysis subspace learning techniques, like LDA, ISOMAP, LLE, LE [18]. Let us denote by $\mathbf{L}_X$ the graph Laplacian matrix describing the discrimination criterion X. Then, the criterion X can be modeled by using a matrix of the form:

$$\mathbf{S}_X = \boldsymbol{\Phi}\mathbf{L}_X\boldsymbol{\Phi}^T. \tag{11}$$

For example, the scatter matrices (8), (9) can be expressed as follows:

$$\mathbf{S}_T = \boldsymbol{\Phi}\mathbf{L}_T\boldsymbol{\Phi}^T, \tag{12}$$

$$\mathbf{S}_w = \boldsymbol{\Phi}\mathbf{L}_w\boldsymbol{\Phi}^T, \tag{13}$$

where the graph Laplacian matrices $\mathbf{L}_T$, $\mathbf{L}_w$ are given by:

$$\mathbf{L}_T = \mathbf{I} - \frac{1}{N}\mathbf{e}\mathbf{e}^T, \tag{14}$$

$$\mathbf{L}_w = \mathbf{I} - \sum_{j=1}^{N_A} \frac{1}{N_j}\mathbf{e}^j\mathbf{e}^{jT}. \tag{15}$$

In (14), (15), $\mathbf{e} \in \mathbb{R}^l$ is a vector of ones, $\mathbf{I} \in \mathbb{R}^{l\times l}$ is the identity matrix and $\mathbf{e}^j \in \mathbb{R}^l$ is a vector with $e_l^j = 1$ if $l = c_i$ and $e_l^j = 0$ otherwise.

Using (10) and (11), it can be shown that $\mathbf{W}_{out}$ can be calculated by:

$$\mathbf{W}_{out} = \left[\mathbf{\Phi}\left(\mathbf{I} + \frac{1}{\lambda_1}\mathbf{L}_X\right)\mathbf{\Phi}^T\right]^{-1}\mathbf{\Phi}\mathbf{T}^T. \tag{16}$$

By substituting the graph Laplacian matrix $\mathbf{L}_X$ in (16) with $\mathbf{L}_T$, $\mathbf{L}_w$, or the graph Laplacian matrices defined for other discrimination criteria, the proposed optimization scheme for $\mathbf{W}_{out}$ calculation can, easily, incorporate various (linear or non-linear) discrimination criteria under the graph embedding framework without any modification.

The above described Discriminant Extreme Learning Machine (DELM) algorithm is able to employ labeled action vectors $\mathbf{s}_i,\ i = 1, \ldots, l$ for SLFN network training. An extension of the algorithm, that is able to exploit information appearing in both labeled and unlabeled action vectors is presented in the following Section.

3.3. Semi-supervised Discriminant Extreme Learning Machine

In this Section, we extend the above described algorithm in order to exploit information appearing in both labeled and unlabeled action vectors for $\mathbf{W}_{out}$ calculation. To this end, we exploit the smoothness assumption of semi-supervised learning [20], where it is expected that if two data samples (action instances) are close to each other, they are likely to share the same class label. That is, by following the notation used in this paper, we would like the action vectors representation in the output space to be close, according to their distance in the ELM space. This can be expressed by minimizing:

$$\sum_{i=1}^{N_I}\sum_{j=1}^{N_I} w_{ij}\left(\mathbf{W}_{out}^T\boldsymbol{\phi}_i - \mathbf{W}_{out}^T\boldsymbol{\phi}_j\right)^2, \tag{17}$$

where $N_I = l + u$ and w_{ij} is a value denoting the similarity between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ in the ELM space.

By expressing the similarity between action vectors in the ELM space using the corresponding graph adjacency matrix $\tilde{\mathbf{B}} \in \mathbb{R}^{N_I \times N_I}$, (17) takes the form $Tr\left(\mathbf{W}_{out}^T \tilde{\mathbf{\Phi}} \tilde{\mathbf{L}} \tilde{\mathbf{\Phi}}^T \mathbf{W}_{out}\right)$, where $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{H \times N_I}$ is a matrix containing the action vectors representation in the ELM space for both the labeled and the unlabeled action vectors, i.e., $\tilde{\mathbf{\Phi}} = [\mathbf{\Phi} | \mathbf{\Phi}_u]$, and $\tilde{\mathbf{L}} \in \mathbb{R}^{N_I \times N_I}$ is the graph Laplacian matrix obtained by using $\tilde{\mathbf{B}}$. By incorporating (17) in (7), $\mathbf{W}_{out}$ can be calculated by minimizing:

$$\mathcal{J}_2 = \frac{1}{2} Tr\left(\mathbf{W}_{out}^T \mathbf{S}_X \mathbf{W}_{out}\right) + \frac{\lambda_1}{2} \|\mathbf{W}_{out}^T \mathbf{\Phi} - \mathbf{T}\|_F^2 + \frac{\lambda_2}{2N_I^2} Tr\left(\mathbf{W}_{out}^T \tilde{\mathbf{\Phi}} \tilde{\mathbf{L}} \tilde{\mathbf{\Phi}}^T \mathbf{W}_{out}\right), \tag{18}$$

where λ_2 is a parameter denoting the importance of the Laplacian regularization in the optimization problem. The term $\frac{1}{N_I^2}$ is used, since it is the natural scale factor for the empirical estimate of the Laplacian operator [20].

By solving for $\frac{\theta \mathcal{J}_2}{\theta \mathbf{W}_{out}} = 0$, $\mathbf{W}_{out}$ is given by:

$$\mathbf{W}_{out} = \left[\tilde{\mathbf{\Phi}} \left(\mathbf{1} + \frac{\lambda_2}{\lambda_1 N_I^2} \tilde{\mathbf{L}}\right) \tilde{\mathbf{\Phi}}^T + \frac{1}{\lambda_1} \mathbf{S}_X\right]^{-1} \mathbf{\Phi} \mathbf{T}^T, \tag{19}$$

where $\mathbf{1} \in \mathbb{R}^{N_I \times N_I}$ is a matrix having $[\mathbf{1}]_{ii} = 1,\ i = 1, \ldots, l$ and $[\mathbf{1}]_{ij} = 0$ otherwise.

In the above discussion, $\tilde{\mathbf{B}}$ can be formed by using various similarity criteria [21, 22]. In the experiments presented in this paper we employ the heat function $e^{-\frac{\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2}{2\sigma^2}}$, where σ is determined to be the mean distance among $\boldsymbol{\phi}_i,\ i = 1, \ldots, N_I$. $\tilde{\mathbf{L}}$ can be calculated either by using a fully connected graph, or by using the K-NN connected graph, i.e., a graph where each action vector is connected only to its K nearest neighbors. Furthermore, while we use the graph Laplacian matrix $\tilde{\mathbf{L}}$ in the above description, the normalized Laplacian matrix $\tilde{\mathcal{L}} = \mathbf{D}^{-1/2} \tilde{\mathbf{L}} \mathbf{D}^{-1/2}$ can be used interchangeably [20]. In fact, we use K-NN ($K = 5$) connected graphs and $\tilde{\mathcal{L}}$ in all our experiments. Finally, similar to $\mathbf{S}_X$, the calculation of $\tilde{L}$ in the ELM space $\mathbb{R}^H$, rather than in the input space $\mathbb{R}^D$, has the advantage that nonlinear relationships, which depend on the adopted activation function $\Phi()$, between training action vectors $\mathbf{s}_i$ can be better described.

While the above described DELM and Semi-supervised Discriminant Extreme Learning Machine (SDELM) classification algorithms can handle the supervised

and the semi-supervised action classification problems, respectively, they can operate only in the cases where each action instance is represented by one action vector $\mathbf{s}_i$. In order to handle the case where each action instance is represented by multiple action vectors $\mathbf{s}_i^v,\ v = 1, \ldots, N_V$ an optimization scheme, called Multi-view Semi-supervised Discriminant Extreme Learning Machine (MSDELM), is described in the following Section.

3.4. Multi-view Semi-supervised Discriminant Extreme Learning Machine

Let us assume that the N_I training (labeled and unlabeled) action instances are represented by the corresponding action vectors $\mathbf{s}_i^v \in \mathbb{R}^{D_v}, i = 1, \ldots, l, \ldots, N_I, v = 1, \ldots, N_V$. We would like to employ them, in order to train N_V SLFN networks, each operating on one view. To this end we map the action vectors of each view v to one ELM space $\mathbb{R}^{H_v}$, by using randomly chosen input weights $\mathbf{W}_{in}^v \in \mathbb{R}^{D_v \times H_v}$ and input layer bias values $\mathbf{b}^v \in \mathbb{R}^{H_v}$. H_v is the dimensionality of the ELM space related to view v.

A commonly acceptable assumption of multi-view semi-supervised learning is that a good classifier can be learned from each view [23]. Therefore, we would expect the outputs of these classifiers to be consistent to each other to a large extent. That is, by following the notation used in this paper, we would like the action vector representations of the same action instance in the output spaces of all the different views to be as close as possible. This can be expressed by minimizing:

$$\sum_{v=1}^{N_V} \sum_{j=1}^{N_V} \sum_{i=1}^{N_I} \|\mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v - \mathbf{W}_{out}^{j\,T} \boldsymbol{\phi}_i^j\|_2^2. \tag{20}$$

By incorporating (20) in (18), $\mathbf{W}_{out}^v$ can be calculated by minimizing:

$$\begin{aligned} \mathcal{J}_3 = \sum_{v=1}^{N_V} \Bigg[\frac{1}{2} Tr\left(\mathbf{W}_{out}^{v\,T} \mathbf{S}_X^v \mathbf{W}_{out}^v\right) \;+\; & \frac{\lambda_1}{2} \|\mathbf{W}_{out}^{v\,T} \boldsymbol{\Phi}^v - \mathbf{T}\|_F^2 + \frac{\lambda_2}{2N_I^2} Tr\left(\mathbf{W}_{out}^{v\,T} \tilde{\boldsymbol{\Phi}}^v \tilde{\mathbf{L}}^v \tilde{\boldsymbol{\Phi}}^{v\,T} \mathbf{W}_{out}^v\right) \\ & + \frac{\lambda_3}{2N_I} \sum_{j=1}^{N_V} \sum_{i=1}^{N_I} \|\mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v - \mathbf{W}_{out}^{j\,T} \boldsymbol{\phi}_i^j\|_2^2 \Bigg], \end{aligned} \tag{21}$$

where $\mathbf{S}_X^v$ and $\tilde{\mathbf{L}}^v$ are the discriminant and graph Laplacian matrices calculated for view v, respectively. Solving for $\frac{\theta \mathcal{J}_3}{\theta \mathbf{W}_{out}^v} = 0$, $\mathbf{W}_{out}^v$ is given by:

$$\mathbf{W}_{out}^v = \left[\tilde{\boldsymbol{\Phi}}^v \left(\mathbf{1} + \frac{\lambda_2}{\lambda_1 N_I^2} \tilde{\mathbf{L}}^v - \frac{\lambda_3 (N_V - 1)}{\lambda_1 N_I} \mathbf{I}\right) \tilde{\boldsymbol{\Phi}}^{v\,T} + \frac{1}{\lambda_1} \mathbf{S}_X^v\right]^{-1} \tilde{\boldsymbol{\Phi}}^v \left(\tilde{\mathbf{T}}^T + \frac{\lambda_3}{\lambda_1 N_I} \sum_{j \neq v} \tilde{\boldsymbol{\Phi}}^{jT} \mathbf{W}_{out}^j\right). \tag{22}$$

Since (22) cannot be directly solved, we propose an iterative optimization scheme for $\mathbf{W}_{out}^v$ calculation. According to this, the output weights of all the views $\mathbf{W}_{out}^v,\ v = 1, \ldots, N_V$ are initiated by using (19). After initializing $\mathbf{W}_{out}^v$, they are iteratively adapted by using:

$$\mathbf{W}_{out,t}^v = \left[\tilde{\mathbf{\Phi}}^v \left(\mathbf{1} + \frac{\lambda_2}{\lambda_1 N_I^2}\tilde{\mathbf{L}}^v + \frac{\lambda_3 N_V}{\lambda_1 N_I}\mathbf{I}\right)\tilde{\mathbf{\Phi}}^{v\,T} + \frac{1}{\lambda_1}\mathbf{S}_X^v\right]^{-1}\tilde{\mathbf{\Phi}}^v\left(\tilde{\mathbf{T}}^T + \frac{\lambda_3}{\lambda_1 N_I}\tilde{\mathbf{O}}_{t-1}^T\right), \tag{23}$$

where $\tilde{\mathbf{O}}_{t-1} = \sum_{j=1}^{N_V}\mathbf{W}_{out,t-1}^{j\,T}\tilde{\mathbf{\Phi}}^j$. Here, we have introduced the index t denoting the iteration of the proposed iterative optimization scheme. The optimization process is terminated when $\sum_{v=1}^{N_V}\|\mathbf{W}_{out}^v(t-1) - \mathbf{W}_{out}^v(t)\|_F \leq \epsilon$, where ϵ is a small positive value, which is set to $\epsilon = 0.01$ in our experiments. In Appendix Appendix A, we show that each step of the proposed iterative optimization scheme is a convex optimization problem having a global solution. Thus, the proposed iterative optimization scheme will converge to a local minimum of $\mathcal{J}_3$ in a finite number of optimization steps.

The above described optimization scheme will result in the adaptation of the output weights of all the SLFN networks, so that their outputs are consistent. However, by using (20), all the SLFN networks, each corresponding to one view, equally contribute to the adaptation process. In the cases where some of the views are not able to train a good classifier, this might hurt the classification performance. Therefore, we would like each of the classifiers to contribute according to its discriminative ability on the adaptation process. To this end, we propose to replace (20) with:

$$\sum_{v=1}^{N_V}\sum_{j=1}^{N_V}\sum_{i=1}^{N_I}\alpha_{j,t}\|\mathbf{W}_{out}^{v\,T}\boldsymbol{\phi}_i^v - \mathbf{W}_{out}^{j\,T}\boldsymbol{\phi}_i^j\|_2^2, \tag{24}$$

where $\alpha_{j,t}$ is a value denoting the discriminative ability of classifier j at iteration t. Exploiting, once again, the fact that the ELM optimization process can be considered to be a data mapping to a low-dimensional space, according to a discriminant analysis perspective, the discriminative ability of each classifier can be calculated by:

$$\alpha_{v,t} = \frac{Tr\left(\mathbf{W}_{out,t}^{v\,T}\mathbf{S}_b^v\mathbf{W}_{out,t}^v\right)}{Tr\left(\mathbf{W}_{out,t}^{v\,T}\mathbf{S}_w^v\mathbf{W}_{out,t}^v\right)}, \tag{25}$$

where $\mathbf{S}_b^v$, $\mathbf{S}_w^v$ are the between-class and within-class scatter matrices calculated

for view v defined as follows:

$$\mathbf{S}_b^v = \sum_{i=1}^{l} \left(\boldsymbol{\phi}_i^v - \bar{\boldsymbol{\phi}}^v\right)\left(\boldsymbol{\phi}_i^v - \bar{\boldsymbol{\phi}}^v\right)^T, \tag{26}$$

$$\mathbf{S}_w^v = \sum_{j=1}^{N_A}\sum_{i=1}^{l} \eta_i^j \left(\boldsymbol{\phi}_i^v - \bar{\boldsymbol{\phi}}_j^v\right)\left(\boldsymbol{\phi}_i^v - \bar{\boldsymbol{\phi}}_j^v\right)^T. \tag{27}$$

By normalizing $\alpha_{v,t}$, in order to have unit l1-norm, i.e., $\sum_{v=1}^{N_V} \alpha_{v,t} = 1$, and incorporating (24) in (21), $\mathbf{W}_{out}^v$ adaptation is performed by:

$$\mathbf{W}_{out,t}^v = \left[\tilde{\boldsymbol{\Phi}}^v \left(\mathbf{1} + \frac{\lambda_2}{\lambda_1 N_I^2}\tilde{\mathbf{L}}^v + \frac{\lambda_3}{\lambda_1 N_I}\mathbf{I}\right)\tilde{\boldsymbol{\Phi}}^{v\,T} + \frac{1}{\lambda_1}\mathbf{S}_X^v\right]^{-1} \tilde{\boldsymbol{\Phi}}^v \left(\tilde{\mathbf{T}}^T + \frac{\lambda_3}{\lambda_1 N_I}\tilde{\mathbf{O}}_{t-1}^T\right), \tag{28}$$

where $\tilde{\mathbf{O}}_{t-1} = \sum_{j=1}^{N_V} \alpha_{j,t-1}\mathbf{W}_{out,t-1}^{jT}\tilde{\boldsymbol{\Phi}}^j$ which is the weighted sum of network outputs corresponding to different views.

In the test phase, a test action instance represented by $N \leq N_V$ test action vectors $\mathbf{s}_t^v,\ v = 1, \ldots, N$ can be introduced to the trained SLFN networks and N network output vectors $\mathbf{o}_t^v,\ v = 1, \ldots, N$ are obtained. The test action instance is classified to the action class corresponding to the maximum mean network output [24]:

$$c_t = arg\max_j \left(\frac{1}{N}\sum_{v=1}^{N} o_{t,j}^v\right),\ \ j = 1, ..., N_A. \tag{29}$$

3.5. Discussion

By observing (16), it can be seen that the ELM algorithm, as well as the ORELM algorithm proposed in [25], are special cases of the proposed DELM algorithm for $\mathbf{S}_X = \mathbf{I}$ and $\mathbf{S}_X = \mathbf{0}$, where $\mathbf{I}$, $\mathbf{0}$, are the identity and zero matrices, respectively. Furthermore, by observing (19), it can be seen that the SELM algorithm [26] is a special case of the proposed SDELM algorithm for $\mathbf{S}_X = \mathbf{0}$. The use of $\mathbf{S}_X = \mathbf{I}$ in (19), would result to an extension of the ORELM algorithm [25] in order to incorporate information coming from both labeled and and unlabeled data. Finally, by using $\mathbf{S}_X^v = \mathbf{0}$, $\mathbf{S}_X^v = \mathbf{0}$ or $\mathbf{S}_X^v = \mathbf{I}$, $\tilde{\mathbf{L}}^v = \mathbf{0}$ in (28), the proposed MVSDELM algorithm can be considered as an extension of the ELM, SELM and ORELM algorithms in order to incorporate information appearing in multiple data views. However, the proposed algorithms are, also, able to incorporate information concerning the training data relationships in the ELM space by exploiting several $\mathbf{S}_X^v$ matrices, as has been discussed in Subsection 3.2.

In order to obtain satisfactory performance by applying the proposed, and all the regularization-based, algorithms appropriate values for the regularization parameters $\lambda_i,\ i = 1, 2, 3$ should be chosen. A common practice, is the determination of the optimal parameter values by applying the cross-validation scheme, following a grid-search strategy. Another parameter that should be determined is the number of hidden layer neurons H. In order to find the optimal ELM space dimensionality several methods have been proposed [27, 28]. Such methods either start by using a large number of hidden neurons and iteratively decrease it as long as the training classification error remains above a pre-defined threshold, or start by using a small number of hidden neurons and iteratively increase it. These methods depend on user pre-specified parameter values, like the maximal number of hidden neurons and acceptable training error threshold. Furthermore, the determined optimal number of hidden neurons depends on the training data at hand. For example, if some of the training data are replaced by others, the optimal number of hidden layer neurons should be determined again. This is why the dimensionality of the ELM space H is usually empirically chosen.

Regarding the calculation of appropriate Graph Laplacian matrices $\tilde{\mathbf{L}}^v$ which are employed for the incorporation of unlabeled data information on the calculation of the networks' output weights, an appropriate number of weights should be exploited for different training samples. Sparsity-based techniques [29] have been found to work well for this purpose, since they are able to automatically determine the number neighboring samples and the corresponding weights for the calculation of appropriate Laplacian matrices. However, such techniques are time-consuming. K-NN connected graphs have been found to work well and this is why they have been widely adopted.

4. Experiments

In this Section, we present experiments conducted in order to evaluate the performance of the proposed action classification algorithms. We present experiments on single-view (semi-)supervised action recognition on the KTH and UCF50 databases in Subsection 4.4. We present experiments on multi-view (semi-)supervised action recognition on the i3DPost and KTH databases in Subsection 4.5, where we investigate the cases of multi-view action recognition by using a multi-camera setup and multi-view action recognition by using multiple action representations, respectively. Comprehensive description of the databases used in our experiments are provided in the following subsections.

We employ the Action Bank [3] and Harris3D STIP detection, followed by HOG/HOF descriptors calculation [4], for action video representation. We compare the performance of the proposed algorithms with that of ELM [12], ORELM [25][1], kernel Support Vector Machine employing RBF kernel (kSVM)[2], kernel Laplacian SVM employing RBF kernel (LapSVM) [20][3], SELM [26] and two-view SVM (SVM2K) [30][4] classifiers. The sigmoid activation function has been used for all the ELM-based classification schemes. The optimal parameter values for all the algorithms have been determined by applying a grid search strategy using the values $C = 10^r$ for ORELM and SELM, $C = 10^r$ and $\sigma = 10^r$ for kSVM, $\gamma_A = 10^r$, $\gamma_I = 10^r$ and $\sigma = 10^r$ for LapSVM and $\lambda_1 = 10^r$, $\lambda_2 = 10^r$, $\lambda_3 = 10^\rho$ for SVM2K and the proposed algorithms for $r = -6, \ldots, 6$, $\rho = -3, \ldots, 3$. Finally, we compare the performance of the proposed action recognition method with that of other methods evaluating their performance on the above-mentioned databases.

4.1. The KTH action database

The KTH action database consists of 600 videos depicting 25 persons, performing six actions each [6]. The actions appearing in the database are: 'walking', 'jogging', 'running', 'boxing', 'hand waving' and 'hand clapping'. Four different scenarios have been recorded: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4), as illustrated Figure 1. The persons are free to change motion speed and direction between different action realizations. The most widely adopted experimental setting on this data set is based on a split (16 training and 9 test persons) that has been used in [6].

4.2. The UCF50 action database

The UCF50 action database consists of 6680 realistic videos taken from YouTube, each belonging to one of 50 action classes. The database is very challenging, due to large variations in camera motion, subject appearance and pose, subject scale, view angle, cluttered background, illumination conditions, etc. For all the 50 categories, the videos are grouped into 25 groups, where each group consists of more than 4 action clips. The video clips in the same group may share some common features, such as the appearance of the same person, similar background,

[1] http://www.ntu.edu.sg/home/egbhuang/elm_codes.html

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[3] http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html

[4] http://www.davidroihardoon.com/Professional/Code.html

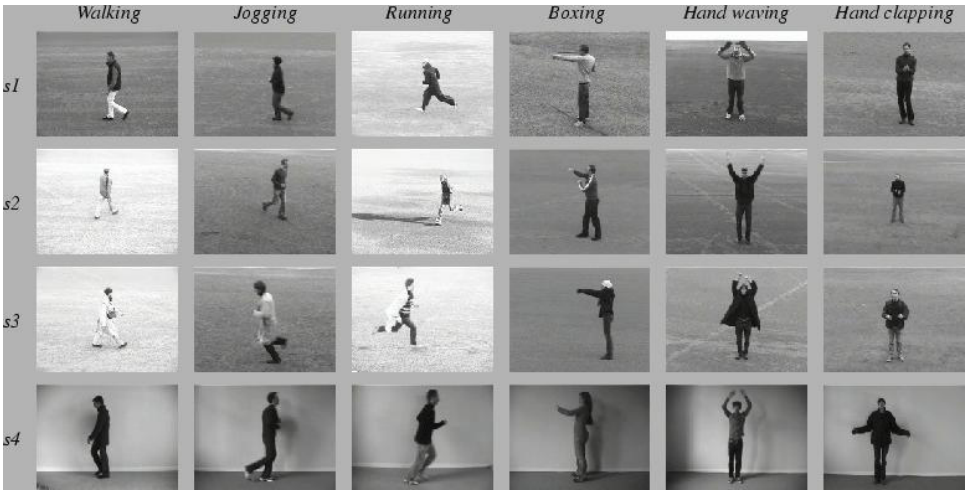

Figure 1: *Video frames from the KTH action database for the four different scenarios.*

similar view angle, and so on. The most widely adopted experimental setting on this database is the 5-fold group-wise cross-validation procedure. That is, the videos are split on five sets, each containing 5 groups. On each fold of the cross-validation procedure, the videos belonging to 4 sets, i.e., 20 groups, are used for training and the videos belonging to the remaining set are used for testing. This procedure is performed 5 times, one for each test set. Example video frames from this database are illustrated in Figure 2.

4.3. The i3DPost action database

The i3DPost multi-view action database [7] contains 512 videos depicting eight persons performing eight actions. The database camera setup consists of eight cameras placed in a ring formation at a height of 2 meters above the studio floor. The actions appearing in the database are: 'walk', 'run', 'jump in place', 'jump forward', 'bend', 'fall down', 'sit on a chair' and 'wave one hand'. The Leave-One-Person-Out cross-validation procedure is used by most action recognition methods evaluating their performance on this data set. That is, the algorithms are trained by using the action videos of seven persons and tested on the videos of the remaining person. Eight evaluation rounds, one for each test person, are performed in order to complete an experiment. Example video frames depicting a person walking as viewed from all eight available view angles are illustrated in Figure 3.

Figure 2: *Video frames from the UCF*50 *action database.*

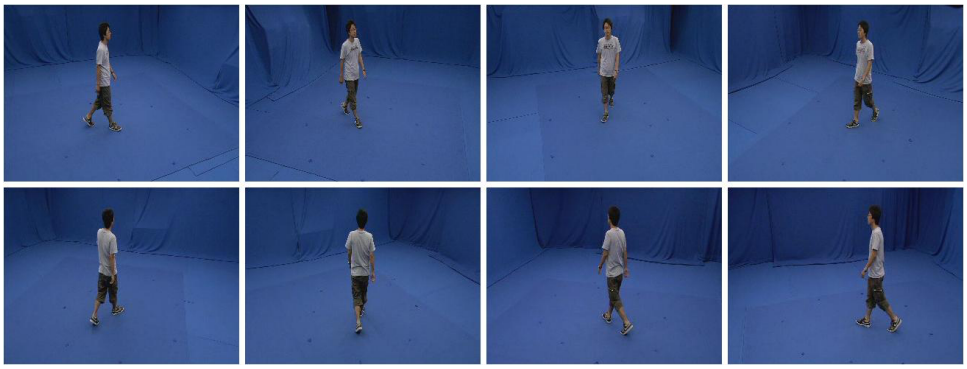

Figure 3: *Video frames from the i3Dpost eight-view action database depicting a person walking.*

4.4. Single-view Action Recognition

In our first set of experiments we have performed supervised action classification on the KTH and UCF50 databases, by employing the Action Bank action video representation. The dimensionality of the 14964-dimensional Action Bank vectors has been reduced by applying PCA, so that 98% of the energy is preserved, resulting to 91- and 467-dimensional feature vectors for the KTH and the UCF50 cases, respectively. The number H of the network hidden layer neurons has been set equal to $H = 500$ and $H = 1000$ for all the ELM-based classification scheme on the KTH and the UCF50 cases, respectively. The mean action classification rates for the ELM, ORELM, kSVM algorithms and the proposed DELM algorithm employing $\mathbf{S}_w$ and $\mathbf{S}_T$ are illustrated in Table 1. As can be seen, the proposed DELM algorithm outperforms all the competing ones in both databases. The corresponding confusion matrices are illustrated in Figures 4 and 5. In Table 1, we also provide the mean training times for all the algorithms. All the experiments have been conducted on a 2.4GHz, 16GB, 64-bit Windows 8 PC, using a MATLAB implementation. As can be seen, the proposed DELM algorithm is computationally efficient, since its learning speed is comparable with that of ELM and ORELM, while the learning process kSVM is quite slow, since it is requires gradient descend based optimization.

In our second set of experiments, we have performed semi-supervised action classification on the KTH and UCF50 databases. We have ordered the training data forming the action classes of the KTH and UCF50 databases by using a random permutation of their indices and used 1% and 5% of them as labeled and the remaining samples as unlabeled data. The action classification rates obtained by following this process and applying the SELM, LapSVM and the proposed

Table 1: Action classification rates and mean training times for supervised classification on the KTH and the UCF50 action databases.

	KTH		UCF50	
	Accuracy	Training Time	Accuracy	Training Time
ELM	90.74%	$89.8ms$	60.6%	$2.3s$
ORELM	99.07%	$98.4ms$	56.28%	$1.3522s$
kSVM	98.15%	$420ms$	57.9%	$30.864s$
DELM ($\mathbf{S}_w$)	98.61%	$192.7ms$	**61.21%**	$1.475s$
DELM ($\mathbf{S}_T$)	**99.54%**	$164.4ms$	60.94%	$1.096s$

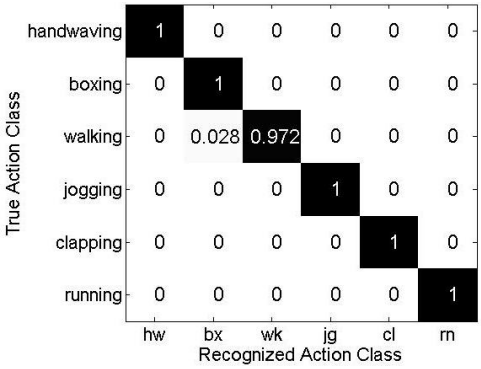

Figure 4: *Confusion matrix for supervised action classification on the KTH database.*

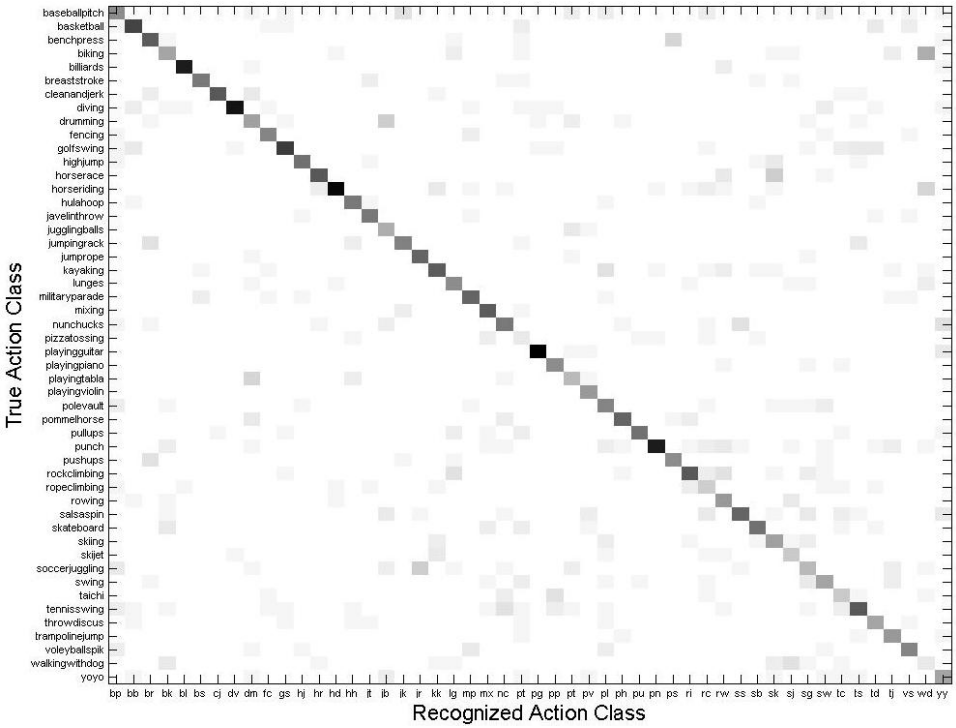

Figure 5: *Confusion matrix for supervised action classification on the UCF*50 *database.*

Table 2: Action classification rates and mean training times for semi-supervised classification on the KTH database.

	KTH			
	$l = 6$ (1 per action class)		$l = 18$ (3 per action class)	
	Accuracy	Training Time	Accuracy	Training Time
SELM	71.76%	$106.7ms$	82.87%	$105.5ms$
LapSVM	**82.41%**	$203.9ms$	**91.2%**	$223.4ms$
SDELM ($\mathbf{I}$)	80.56%	$115.1ms$	90.74%	$116.8ms$
SDELM ($\mathbf{S}_w$)	80.09%	$134.9ms$	90.74%	$145.1ms$
SDELM ($\mathbf{S}_T$)	77.31%	$126.6ms$	**91.2%**	$137ms$

Table 3: Action classification rates and mean training times for semi-supervised classification on the UCF50 database.

	UCF50			
	$l = 0.01N_I$		$l = 0.05N_I$	
	Accuracy	Training Time	Accuracy	Training Time
SELM	11.25%	$4.7824s$	17.01%	$4.8281s$
LapSVM	14.43%	$30.8646s$	31.54%	$17.9869s$
SDELM ($\mathbf{I}$)	14.38%	$3.6857s$	32.2%	$3.7191s$
SDELM ($\mathbf{S}_w$)	**16.54%**	$1.5262s$	32.12%	$1.7585s$
SDELM ($\mathbf{S}_T$)	16.5%	$1.3159s$	**33.12%**	$1.3091s$

SDELM algorithms employing $\mathbf{I}$, $\mathbf{S}_w$ and $\mathbf{S}_T$ are illustrated in Tables 2,3. As can be seen, the proposed SDELM algorithm outperforms both the SELM and the LapSVM algorithms in most cases. Furthermore, it can be seen that the ELM-based classification schemes are computationally more efficient compared to the LapSVM algorithm. The confusion matrix obtained by applying the proposed SDELM algorithm, employing $\mathbf{S}_T$, on the KTH database for the case of $l = 0.05N_I$ is illustrated in Figure 6. The confusion observed between the actions boxing and walking can be explained by the holistic nature of Action Bank action video representation [3].

4.5. Multi-view Action Recognition

As it has been already mentioned, multi-view action recognition refers to the cases where an action instance is represented by multiple action representations either by using multiple action descriptors, or by using a multi-camera setup. In

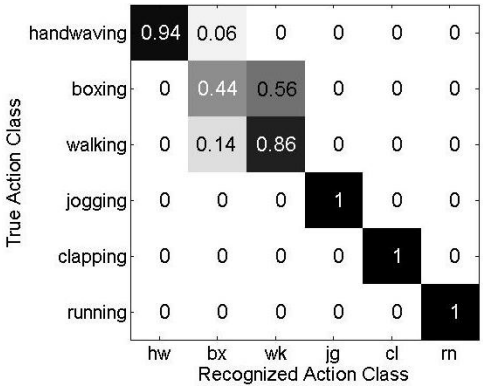

Figure 6: *Confusion matrix for semi-supervised action classification on the KTH database ($l = 0.05N_I$).*

Table 4: Action classification rates for multi-view semi-supervised classification on the KTH database.

	$l = 6$ (1 per action class)			$l = 18$ (3 per action class)		
	HOG	HOF	Multi-view	HOG	HOF	Multi-view
SVM2K	53.24%	80.09%	74.07%	55.09%	78.7%	78.24%
MVSDELM ($\mathbf{I}$)	52.31%	63.43%	**82.41%**	64.35%	77.78%	**81.94%**
MVSDELM ($\mathbf{S}_w$)	52.31%	63.45%	**82.41%**	61.57%	75.66%	74.54%
MVSDELM ($\mathbf{S}_T$)	32.87%	27.78%	79.63%	61.57%	75.66%	75%

order to perform multi-view action recognition by using multiple action descriptors, we have employed the Harris3D detector to detect STIPs on the action videos of the KTH database. HOG and HOF descriptors have been calculated at STIP video locations and the Bag of Features (BoFs)-based action video representation for each descriptor has been calculated. The dimensionality of the obtained action representations has been determined to be equal to $D_v = 300,\ v = 1, 2$. We have used the HOG- and HOF-based action video representations as two views of each action video and performed two-view semi-supervised action classification by applying the SVM2K and the proposed MVSDELM algorithms using $H = 1000$ newtork hidden layer neurons. The obtained action classification rates are illustrated in Tables 4,5.

As can be seen in Tables 4,5, the use of multiple views provides enhanced classification performance, compared to that obtained for each view when used

Table 5: Action classification rates for multi-view semi-supervised classification on the KTH database.

	$l = N_I$		
	HOG	HOF	Multi-view
SVM2K	71.3%	93.8%	93.52%
MVSDELM (**I**)	73.15%	91.67%	94.91%
MVSDELM ($\mathbf{S}_w$)	72.69%	91.2%	**95.37%**
MVSDELM ($\mathbf{S}_T$)	71.3%	90.74%	94.91%

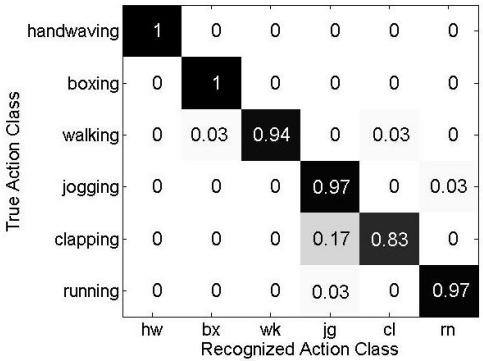

Figure 7: *Confusion matrix for multi-view action classification on the KTH database.*

independently. HOF-based action video representation seems to be more discriminative compared to the HOG-based one, since it consistently provides higher action classification rates. The action classification rates obtained by using the multi-view approach and the proposed MVSDELM algorithm are higher than the ones obtained by using either the HOG- or the HOF-based action representations in most cases, while this is not the case for the SVM2K algorithm. By using the labeling information of 1% of the training action videos (corresponding to $l = 6$) and the proposed MVSDELM algorithm, an action classification rate equal to 82.41% has been obtained. This classification rate is increased to 95.37% when exploiting the labeling information of all the training action videos ($l = N_I$). The confusion matrices obtained by applying the proposed MVSDELM algorithm exploiting $\mathbf{S}_w$ for the supervised and the 1% semi-supervised cases, are illustrated in Figures 7 and 8, respectively.

In order to perform multi-view semi-supervised action recognition by using multiple action representations obtained by using a multi-camera setup, we conducted experiments on the i3DPost eight-view database. We have employed the

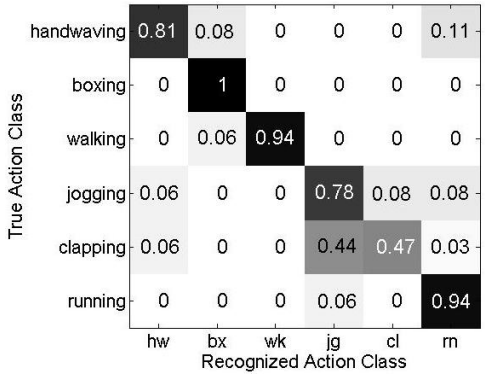

Figure 8: *Confusion matrix for multi-view semi-supervised action classification on the KTH database ($l = 0.01N_I$).*

Table 6: Action classification rates for multi-view semi-supervised classification on the i3DPost database.

	$L = 8$ (1 per action class)	$L = 16$ (2 per action class)	$L = N_I$
MVDELM ($\mathbf{I}$)	12.5%	84.38%	**100%**
MVDELM ($\mathbf{S}_w$)	12.5%	85.94%	98.44%
MVDELM ($\mathbf{S}_T$)	12.5%	85.94%	98.44%
MVSDELM ($\mathbf{I}$)	14.06%	85.94%	96.88%
MVSDELM ($\mathbf{S}_w$)	**51.56%**	**89.06%**	98.44%
MVSDELM ($\mathbf{S}_T$)	31.31%	**89.06%**	98.44%

Harris3D detector to detect STIPs on the action videos and calculated HOG and HOF descriptors on STIP locations. The the BoFs-based action video representation has been calculated by employing concatenated HOG/HOF descriptors. The action videos corresponding to each view angle, with respect to the human body orientation, have been used in order to determine eight views of the performed actions. The dimensionality of the adopted action representations has been determined to be equal to $D_v = 150,\ v = 1, \ldots, 8$ and a number of $H_v = 500$ network hidden layer neurons has been used for all the views. It should be noted here that, in this case, while view determination can be manually performed in the training phase, a viewing angle identification process should be performed in the test phase for automatic view determination [8]. The obtained action classification rates for different numbers of labeled action instances are illustrated in Table 6.

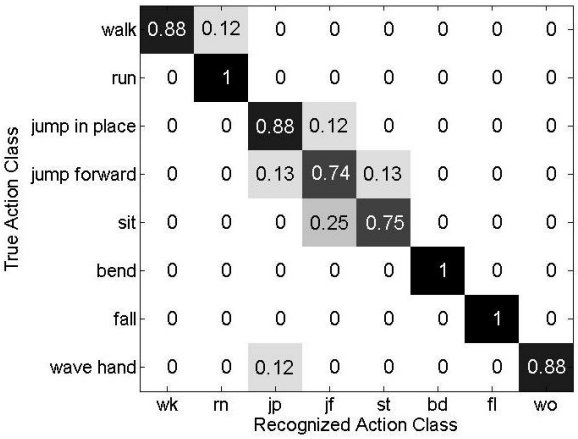

Figure 9: *Confusion matrix for multi-view semi-supervised action classification on the i3DPost database ($L = 16$).*

As can be seen, by using one labeled action instance for each action class ($L = 8$), MVDELM fails to well discriminate the action classes providing an action classification rate equal to 12.5%. MVSDELM employing $\mathbf{S}_T$ is able to better discriminate action classes providing an action classification rate equal to 31.31%, while MVSDELM employing $\mathbf{S}_w$ outperforms all the algorithms providing an action classification rate equal to 51.56%. In the case where two labeled action instances for each action class are employed ($L = 16$), the MVDELM algorithm is able to better discriminate action classes providing action classification rates equal to 84.38% and 85.94% for the cases of $\mathbf{I}$ and $\mathbf{S}_w$, $\mathbf{S}_T$, respectively. MVSDELM employing both $\mathbf{S}_w$, $\mathbf{S}_T$ outperforms MVDELM by providing an action classification rate equal to 89.06%. The confusion matrix of this experiment is illustrated in Figure 9. Finally, by exploiting the action class labels corresponding to all the training action instances ($L = N_I$), MVDELM algorithm is able to perfectly classify all the test action instances providing an action classification rate equal to 100%.

4.6. Comparison with state-of-the-art

In Tables 7, 8, 9 we compare the performance of the proposed action recognition methods with that of other methods evaluating their performance on the KTH, UCF50 and i3DPost databases, respectively. As can be seen, the proposed classification algorithms provide state-of-the-art performance on all the three databases. Specifically, it can be seen that in the KTH database, the proposed MVSDELM

Table 7: Comparison results on the KTH database.

	Action Representation	Accuracy
Method [31]	low-level	84.3%
Method [32]	low-level	86.8%
Method [33]	low-level	91.1%
Method [34]	low-level	91.6%
Method [35]	low-level	91.8%
Method [36]	low-level	93.2%
Method [37]	low-level	93.8%
Method [38]	low-level	93.9%
Method [39]	high-level	94.3%
Method [40]	high-level	94.5%
Method [41]	high-level	94.5%
Method [42]	high-level	94.5%
Method [3]	high-level	98.2%
MVSDELM ($\mathbf{S}_w$)	low-level	**95.37%**
DELM ($\mathbf{S}_T$)	high-level	**99.54%**

algorithm combined with a low-level action representation provides an action classification rate equal to 95.37%, which is 1.47% higher than the best reported action classification rate for low-level action representations. Furthermore, by adopting a high-level action representation, the proposed DELM algorithm provides an action classification rate equal to 99.54%, which is 1.34% higher than the classification rates reported for high-level action representations. In the UCF50 database, the proposed DELM algorithm provides an action classification rate equal to 61.21%, which is 3.31% higher than the best reported action classification performance reported for this data set. Finally, in the i3DPost database, the proposed MVDELM algorithm was able to perfectly classify all the test action instances providing an action classification rate equal to 100%, which is highest reported performance on this dataset.

5. Conclusions

In this paper, we proposed three novel classification algorithms aiming at (semi-)supervised action classification. Discrimination criteria are incorporated to the ELM optimization process in order to enhance the performance of the ELM network. Proper regularization terms are incorporated in the ELM optimization

Table 8: Comparison results on the UCF50 database.

	Accuracy
Method [43]	38.8%
Method [4]	47.9%
Method [3]	57.9%
DELM ($\mathbf{S}_T$)	**61.31%**

Table 9: Comparison results on the i3DPost database.

	Accuracy
Method [44]	92.19%
Method [45]	94.34%
Method [46]	94.37%
Method [8]	94.87%
Method [15]	95.5%
Method [47]	96.34%
Method [48]	98.44%
Method [16]	100%
MVDELM (**I**)	**100%**

process in order to extend the ELM algorithm to multi-view semi-supervised action classification. The proposed algorithms have been evaluated on human action recognition providing state-of-the-art performance on three publicly available databases.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART).

Appendix A.

Here we show that each step of the iterative optimization process employed for $\mathcal{J}_3$ minimization corresponds to a convex optimization problem. By using (21), the derivative $\frac{\vartheta \mathcal{J}_3}{\vartheta \mathbf{W}_{out}^v}$ is given by:

$$\begin{aligned}\frac{\vartheta \mathcal{J}_3}{\vartheta \mathbf{W}_{out}^v} &= \left(\mathbf{S}_X^v + \lambda_1 \mathbf{\Phi}^v \mathbf{\Phi}^{v\,T} + \frac{\lambda_2}{N_I^2}\tilde{\mathbf{\Phi}}^v \tilde{\mathbf{L}}^v \tilde{\mathbf{\Phi}}^{v\,T} + \frac{\lambda_3 (N_V - 1)}{N_I}\tilde{\mathbf{\Phi}}^v \tilde{\mathbf{\Phi}}^{v\,T}\right)\mathbf{W}_{out}^v \\ &- \lambda_1 \mathbf{\Phi}^v \mathbf{T}^T - \frac{\lambda_3}{N_I}\sum_{j \neq v} \tilde{\mathbf{\Phi}}^v \tilde{\mathbf{\Phi}}^{j\,T} \mathbf{W}_{out}^j \\ &= \mathbf{A}^v \mathbf{W}_{out}^v - \lambda_1 \mathbf{\Phi}^v \mathbf{T}^T - \frac{\lambda_3}{N_I}\sum_{j \neq v} \tilde{\mathbf{\Phi}}^v \tilde{\mathbf{\Phi}}^{j\,T} \mathbf{W}_{out}^j, \end{aligned} \tag{A.1}$$

where:

$$\begin{aligned}\mathbf{A}^v &= \mathbf{S}_X^v + \lambda_1 \mathbf{\Phi}^v \mathbf{\Phi}^{v\,T} + \frac{\lambda_2}{N_I^2}\tilde{\mathbf{\Phi}}^v \tilde{\mathbf{L}}^v \tilde{\mathbf{\Phi}}^{v\,T} + \frac{\lambda_3 (N_V - 1)}{N_I}\tilde{\mathbf{\Phi}}^v \tilde{\mathbf{\Phi}}^{v\,T} \\ &= \tilde{\mathbf{\Phi}}^v \left(\mathbf{\Lambda}^v + \lambda_1 \mathbf{1} + \frac{\lambda_2}{N_I^2}\tilde{\mathbf{L}}^v + \frac{\lambda_3 (N_V - 1)}{N_I}\mathbf{I}\right)\tilde{\mathbf{\Phi}}^{v\,T} = \tilde{\mathbf{\Phi}}^v \tilde{\mathbf{A}}^v \tilde{\mathbf{\Phi}}^{v\,T}, \end{aligned} \tag{A.2}$$

In (A.2), $\mathbf{\Lambda}^v = \begin{bmatrix} \mathbf{L}^v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N_i \times N_I}$, where $\mathbf{0}$ is a matrix of zeros with appropriate dimensions. By using (A.1), we obtain $\frac{\vartheta^2 \mathcal{J}_3}{\vartheta \mathbf{W}_{out}^{v\,2}} = \mathbf{A}^v$. From (A.2) it is straightforward to see that the matrix $\tilde{\mathbf{A}}^v$ is positive definite. Since $\tilde{\mathbf{A}}^v$ is positive definite, it can be decomposed as $\tilde{\mathbf{A}}^v = \mathbf{B}\mathbf{B}^T$. From (A.2), $\mathbf{A}^v$ is positive semi-definite for $N < H$ and $\mathbf{A}^v$ is positive definite for $N \geq H$ (which is usually

the case of semi-supervised learning). Thus, $\mathcal{J}_3$ is a convex optimization problem with respect to $\mathbf{W}_{out}^v$ and its global minimum can be obtained by solving for $\frac{\vartheta \mathcal{J}_3}{\vartheta \mathbf{W}_{out}^v} = 0$ and is given by (22).

References

[1] H. Seo, P. Milanfar, Action Recognition from One Example, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 867–882.

[2] I. Laptev, On space-time interest points, International Journal of Computer Vision 64 (2) (2005) 107–123.

[3] S. Sadanand, J. Corso, Action Bank: A High-Level Representation of Activity in Video, IEEE Conference on Computer Vision and Pattern Recognition (2012) 1–8.

[4] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, British Machine Vision Conference 42 (1) (2009) 1–11.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2247–2253.

[6] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: A local SVM approach, International Conference on Pattern Recognition (2004) 32–36.

[7] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, I. Pitas, The i3DPost multi-view and 3D human action/interaction database, Conference on Visual Media Production (2009) 159–168.

[8] A. Iosifidis, A. Tefas, I. Pitas, View-Invariant Action Recognition Based on Artificial Neural Networks, IEEE Transactions on Neural Networks and Learning Systems 23 (3) (2012) 412–425.

[9] X. Y. L. H. Y. Yang, S. Wang, L. Jiao, Semi-supervised action recognition in video via Labeled Kernel Sparse Coding and sparse L1 graph, Pattern Recognition Letters 33 (2012) 1951–1956.

[10] X. P. C. Zhao, X. Li, S. Wang, Human action recognition based on semi-supervised discriminant analysis with globalconstraint, Neurocomputing .

[11] S. X. C. Zhang, T. Liu, H. Lu, Boosted multi-class semi-supervised learning for human action recognition, Pattern Recognition 44 (2011) 2334–2342.

[12] G. Huang, Q. Zhu, C. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, IEEE International Joint Conference on Neural Networks, 2004. Proceedings 2 (2004) 985–990.

[13] P. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, IEEE Transactions on Information Theory 44 (2) (1998) 525–536.

[14] R. Minhas, A. Baradarani, S. Seifzadeh, Q. Jonathan Wu, Human action recognition using extreme learning machine based on visual vocabularies, Neurocomputing 73 (10-12) (2010) 1906–1917.

[15] A. Iosifidis, A. Tefas, I. Pitas, Multi-view Human Action Recognition Under Occlusion based on Fuzzy Distances and Neural Networks, European Signal Processing Conference .

[16] A. Iosifidis, A. Tefas, I. Pitas, Minimum Class Variance Extreme Learning Machine for Human Action Recognition, IEEE Transactions on Circuits and Systems for Video Technology 23 (1) (2013) 1968–1979.

[17] A. Iosifidis, A. Tefas, I. Pitas, Dynamic action recognition based on Dynemes and Extreme Learning Machine, Pattern Recognition Letters, 34 (2013) 1890–1898.

[18] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.

[19] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd ed, Wiley-Interscience, 2000.

[20] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7 (2006) 2399–2434.

[21] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Advances in neural information processing systems 14 (2001) 585–591.

[22] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[23] S. Sun, Multi-view Laplacian support vector machines, International Conference on Advanced Data Mining and Applications (2011) 209–222.

[24] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (3) (1998) 226–239.

[25] G. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42 (2) (2012) 513–529.

[26] J. Liu, Y. Chen, M. Liu, Z. Zhao, SELM: Semi-supervised ELM with application in sparse calibrated location estimation .

[27] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, IEEE Transactions on Neural Networks 21 (1) (2010) 158–162.

[28] G. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16-18) (2007) 3056–3062.

[29] A. Iosifidis, A. Tefas, I. Pitas, Learning sparse representations for view-independent human action recognition based on fuzzy distances, Neurocomputing 121 (2013) 334–353.

[30] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor, S. Szedmak, Two View Learning: SVM-2K, Theory and Practice, Conference on Neural Information Processing Systems (NIPS) .

[31] A. Klaser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, British Machine Vision Conference (2009) 995–1004.

[32] S. Savarese, A. Delpozo, J. Niebles, L. Fei-Fei, Spatial-temporal correlations for unsupervised action classification, IEEE Motion and Video Computing (2008) 1–8.

[33] M. Ryoo, M. Shah, J. Aggarwal, Spatio-temporal relationship matdch: Video structure comparison for recognition of complex huma activities, IEEE Computer Vision and Pattern Recognition (2009) 1593–1600.

[34] J. Liu, B. Kuipers, S. Savarese, Recognizing human actions by attributes, IEEE Computer Vision and Pattern Recognition (2011) 3337–3344.

[35] I. Laptev, M. Marszalek, C. Schmind, B. Rozenfeld, Learning realistic human actions from movies, IEEE Computer Vision and Pattern Recognition (2008) 1–8.

[36] M. Bregonzio, S. Gong, T. Ziang, Recognizing action as clouds of space-time interest points, IEEE Computer Vision and Pattern Recognition (2009) 1948–1955.

[37] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos, IEEE Conference on Computer Vision and Pattern Recognition (2009) 1996–2003.

[38] Q. Le, W. Zou, S. Yeung, A. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, IEEE Conference on Computer Vision and Pattern Recognition (2011) 3361–3368.

[39] J. Liu, M. Shah, Learning human actions via information maximization, IEEE Conference on Computer Vision and Pattern Recognition (2008) 1–8.

[40] A. Gilbert, J. Illingworth, R. Bowden, Fast realistic multi-action recognition using mined dense spatio-temporal features, International Conference on Computer Vision (2009) 925–931.

[41] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, IEEE Conference on Computer Vision and Pattern Recognition (2010) 2046–2053.

[42] X. Wu, D. Xu, L. Duan, J. Luo, Action recognition using context and appearance distribution features, IEEE Conference on Computer Vision and Pattern Recognition (2011) 489–496.

[43] A. Olivia, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (2001) 145–175.

[44] M. Holte, T. Moeslund, N. Nikolaidis, I. Pitas, 3D Human Action Recognition for Multi-View Camera Systems, IEEE International Conference on

3D Imaging, Modeling, Processing, Visualization and Transmission (2011) 342–349.

[45] A. Iosifidis, A. Tefas, N. Nikolaidis, I. Pitas, Multi-view human movement recognition based on Fuzzy distances and Linear Discriminant Analysis, Computer Vision and Image Understanding 116 (3) (2012) 347–360.

[46] A. Iosifidis, A. Tefas, I. Pitas, Neural representation and learning for multi-view human action recognition, IEEE International Joint Conference on Neural Networks (2012) 2233–2238.

[47] A. Iosifidis, A. Tefas, I. Pitas, Multi-view Action Recognition Based on Action Volumes, Fuzzy Distances and Cluster Discriminant Analysis, Pattern Recognition Letters .

[48] M. Holte, B. Chakraborty, J. Gonzalez, T. Moeslund, A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points, IEEE Journal of Selected Topics in Signal Processing 99 (2012) 553–565.